Back to the Future – Pollock versus Toulmin

Ray Feodoroff
MSc Safety, Risk and Reliability Engineering
Raytheon Australia PTY LTD

Abstract

Is the aim to better integrate safety engineering practices at all too late? Is there something about basing argumentation notations on Toulminesque principles that has forestalled integration of assurance argumentation into system designs? How might Pollock have influenced Argumentation notations? Is there a resignation in the industry, keeping a stonewall between design and argument? Grand assurance standards such as OMG SACM(ARM) and IEEE/IEC/ISO 15026[1] suggest so.

The most visible aspect of safety engineering appears to be the outcomes of the safety case in the form of the assurance arguments. These safety case arguments tend to be poorly integrated with the system design, and so create an assurance problem in the potential for breakdown of traceability between artefacts – due to the disparities between the underlying structure of the conceptual representations within the separate design and safety case artefacts.

This paper investigates, via a literature survey and case study, some of the historical aspects of the evolution of argumentation notations to see how the choice of semantics of those notations may have pushed safety out of design. Taking the cue from discussion around defeasible reasoning in the argumentation notation literature, this paper will look at semantics that may be injectable into design notations that could see reasoning about safety concerns integral to the design notation.

1 Introduction

Proponents of Goal Structure Notation (GSN) adopted the Toulminesque principle of a typed argument framework when evolving the notation to the standard in general use per Kelly (1998, p. 62) - this despite rejecting Toulminesque reasoning as being too general. The influence GSN has in the safety justification domain comes from it being a typed argument framework. This is despite there being no means for GSN to meet the second criteria Kelly claims, that is, to explicitly capture concepts that relate to the safety domain (such as system models). What is also missing from the GSN notation is the semantics of scepticism.

Notions of epistemic reasoning, and its application to safety cases, has framed some of the debate (Rushby 2013, Holloway and Johnson 2009, Graydon and Holloway 2015). As GSN is a structured notation it gains epistemic qualities - though fallacy still counteracts the benefits of the structure as fallacy also comes from the structure per Greenwell et al. (2006). Regardless, the notion that the argument is structured aims to counter scepticism since it lays bare the logic. The logic, however, has sensitivity to what is unknown and to doubt. Detection of errors in inference, evidence and in the accuracy of goals can raise doubts per Duan et al. (2017). Argumentors feel that the expression of confidence will act to counteract doubt.

Duan et al. warn of the downfall of separate arguments for confidence in safety cases, given the potential for a large incomprehensible confidence case, alongside a large incomprehensible assurance case. That is, they discount application of even more argument. In concert, Hawkins et al. (2011) aim to stem the propensity to over-argue, by way of the introduction of Assurance Claim Points (ACPs). ACPs act as a Baconian mechanism to express confidence over the GSN graph, by decorating the graph with assertions of sufficiency of satisfaction. ACPs, thereby, try to touch the graph with inductive reasoning, by trying to weight the graph edges.

According to Duan et al, at the other end of the confidence mechanism scale, Zhao et al. (2012) reprise the role of Toulminesque reasoning in review of arguments when applying a Pascalian mechanism to argue twice – via epistemic assurance and then again via aleatoric confidence by use of Bayesian Belief Networks (BBN).

There are other attempts at injection of confidence, but the upshot is that confidence comes from a balance of evidence that both supports and detracts from a claim as expressed. That is, confidence in the satisfaction of a claim (or lack of confidence) is the net result of the application of scepticism.

The problem, caused by epistemic standards, is that the throw out generally to safety case notations, as the necessary mechanism for capturing argumentation, is fraught. The discussion around epistemic standards leads to the need to consider that one can likely ignore application of, or defence against, scepticism in epistemic low-standard, tending towards monotonic, contexts. The absence of documented scepticism in the argument, however, does not mean the epistemic context is of a low-standard. To wit, the absence of semantic mechanisms for depicting scepticism within a notation seems to better fit contexts addressable by epistemic low-standards. In the case of epistemic high-standards, the injection of doubt may be a side effect of the use of a semantically challenged notation – as this may tend to inject information loss and therefore also challenge comprehension (cf. Hawkins et al. 2011; Zhao et al. 2012; Armstrong and Paynter, 2004).

Hence, there is discussion in the argumentation literature in the role of scepticism, expressed as discussion around Anti-Goals or more generally of Pollock's defeasible reasoning (Armstrong and Paynter 2004, Rushby et al. 2015, Goodenough et al. 2015). Defeasible reasoning is thus necessary to foster confidence in the argument. This overturns the original denouncement of Toulmin's "notation" (Kelly 1998, p 62), as Toulmin's "notation"[2] included the notion of exception or rebuttal

[1] These standards now dictate the semantics for assurance argumentation notations!

[2] Note Kelly refers to Toulmin's "notation". Toulmin does not refer to a notation. Rather, Toulmin refers to the layout of arguments. Notations might, indeed, need more work on their semantics.

(Toulmin 2003). This quality of defeasibility therefore also now acts to frame the criteria used when choosing typed argument frameworks in the future.

All this debate leads one to the conclusion, however, that there is a tension in the argumentation notation camp - between notations, and between notations and the raw philosophy of argumentation. How then do practical reasoners, in the requirements and design space, approach contributing to assurance argumentation, short of becoming philosophers?

2 Goal couching and verb tense

As per GSN, goal-oriented requirement notations (GORN) use goals as a syntactic element. As per GSN, GORN can interpret goals as claims since that appears to be the propensity. The difference is subtle in any event, relating to the couching of the goals.

The verb-noun pairing of GSN goal declarations is a similar requirement to that of GORN goals. Within goal-oriented requirement engineering (GORE), goal verb-noun pairing spans the classes of action-verbs, being:

- Attainment
- Cessation
- Avoidance, and
- Maintenance.

Generally, couching of GORN goals is in the imperative intent, namely:

- attain,
- cease,
- avoid, and
- maintain.

GSN goals are to act as propositions since they lead to claims (Kelly, 1998, p. 86). The notion of the GSN goal as a proposition, that can either assert or deny, likely incurs a need for an agreement on the speech act taxonomy. For example, deny(X) is syntactic sugar for assert(not(X)). In that vein, the notion of assert/deny is akin to expressing goals as Attainment versus Cessation, or Maintenance versus Avoidance[3].

In fact, the difference in goal couching between the rationale and argument camps is only in the tense[4]. As they aim to assert success, the GSN goals then span the past participles of GORE action-verb classes, namely:

- attained,
- ceased,
- avoided, and
- maintained.

There are, therefore, no syntactic or semantic barriers prohibiting couching GORN goals as past participles. GSN proponents have, indeed, opted to use the imperative form of the verb when using GSN in place of GORN (Habli et al. 2007).

That is, the tense of the verb is not sacrosanct, and it only allows expression of either the anterior or posterior temporal reasoning tense. This relates to the system lifecycle where the anterior sense is aspirational when used during concept and requirements development, and the posterior sense when used during qualification of the system and therefore it's goals.

Incongruous then is that debate over verb tense was had in the GSN camp (Kelly, 1998, pp. 203-204). The resolution was to opt for the past participle form, as a convention, and then acknowledge the optional use of the imperative form. The optional use for the imperative verb form in GSN is, however, the declaration of aspiration that is the anti-pattern of argument per Kelly. The choice of the imperative verb form, taken as the basis for criticism of GORN (Kelly 1998, p. 65), somehow does not act as a criticism when chosen for GSN[5] (cf. Habli et al. 2007).

Aspiration occurs thrice then in GSN usage: the optional verb tense in the imperative form; the temporal tense of the early phases of a phased safety case; and the positional tense of the top goal of an argument tree, before meandering down to its leaves. Aspiration is otherwise the Argumentors' equivocation of the goal-oriented notion of Intention, or as Leveson would refer, Intent[6].

The principle difference between GSN and GORN, in relation to semantics to support non-demonstrative reasoning, is that GORN are notations for design rationale capture and requirement elicitation and will often include weighted reasoning. Fortuitous because both inductive and defeasible reasoning will have "weighted" edges in their graphs. This report will describe the concepts using Goal-oriented Requirement Language (GRL). GRL is a part of ITU-T Z.151 User Requirement Notation (URN)[7,8].

3 Of non-monotonic scenarios

Argumentors have tended to use defeasible reasoning outside of GSN as it is foreign to the semantics of the notation. The alternative to reactive patching of GSN, is to look at logics and notations that might address the push for defeasible reasoning within safety case arguments. In this way, vetting for alternate notations to GSN may suffice to avoid hacking and patching (cf. Assurance Claim Points).

Indeed, there is likely a heuristic, as is applied to software and as likely applies here, that if more than 30% change need take place then one should start afresh. Would the inclusion of defeasible semantics, lexicon and syntax to GSN lead to greater than a 30% change? What other semantics or formalisms are needed to accommodate better integration of defeasible safety reasoning into the system design approaches? To allay fears of starting the notation design process afresh, it will be argued here that the need for defeasible reasoning now acts as a counter argument against the original (Kelly 1998, p. 65) and

---

[3] Note the temporal nature in the pairings. These relates to the span of time.

[4] This relates to the pre/post-condition or state.

[5] And since both counter-intuitions came from the same source.

[6] There is a cross-over between MIT and York at this point. The bridge is cognitive-science.

[7] https://www.itu.int/rec/T-REC-Z.151/en

[8] Think of IEEE/IEC/ISO 15026 as ITU-T Z.151 minus agency, defeasible reasoning and causal thread formalisms.

subsequent stances Argumentors took against GORN (Habli et al. 2007, Wu 2007, Hall-May and Kelly 2011).

This section will now interpret GRL in the context of Backing-Undercutting Argumentation Frameworks (BUAFs). BUAFs give a concrete approach to represent argumentative or non-monotonic scenarios where information can be both attacked and/or supported in the defeasible reasoning sense. The aim of this report is to provide evidence GRL can capture and represent defeasible inferences and thereby is in-line with the current discussion in the GSN literature about the necessity for defeasible reasoning. Mapping GRL to BUAF concepts will therefore justify GRL as an argumentation notation for defeasible reasoning over the system requirements and quality attributes. Qualification of the system requirements and design decisions then becomes the basis for confidence in the attainment of assurance goals.

3.1 BUAF in a nutshell

Dung (1995) investigated the role of acceptability of arguments in non-monotonic reasoning, principally for its application in AI. To that end Dung devised an Abstract Argumentation Framework (AF). Non-monotonic reasoning is a class of logics created to capture defeasible inferences. For a precis of Dung see Cohan et al. (2011) who depict Dung's model by way of a graph of attacking claims (see Figure 1, → = Attacking).

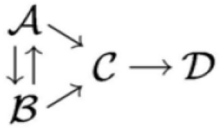

Figure 1: Map of attacking of claims

Monotonic reasoning means that in the face of more weakening premises the claim holds. Given its semantics and syntax, GSN appears limited then to making monotonic claims. This is because GSN lacks semantics within the notation for attack. So, the semantics of GSN does not encourage the application of weakening and likely leads to a presumption of its absence.

Certainly, the propensity to rely on the 'silent' AND by way of part-whole decomposition of claims lends itself to a fulminatory nature – ignoring everything not claimed and denouncing counter-claims.

Ultimately, however, if the requirement is for arguing over non-demonstrable outcomes, then one recognizes that abstract argumentation frameworks, that are non-monotonic in nature, can set the scene for testing logics and thereby the notations applied when reasoning.

Cohen et al. resolved to take the notion of AF by Dung and harmonize it with both Toulmin and Pollock to balance the ideas of rebuttal and undercutting defeaters. The basic notion Cohen et al. propose is that there are three forms of attack: rebuttal, undermining and undercutting. Rebuttal is attacking the main claim directly while undermining otherwise attacks the underlying claims. Undercutting defeats utterly. Undermining balances attacking versus backing to decide the outcome.

Below in Figure 2, the notation Cohen et al. present uses a backing and an undermining edge. Figure 2 thus acts as a key for the basic notation. Cohen et al. also include a preferences notation.

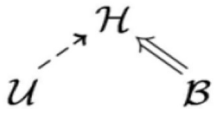

Figure 2: Key: Undermining versus Backing

For example, a preference relationship relates two claims A and B. The notation $\mathcal{A} \preceq \mathcal{B}$ means argument B is at least as preferred as argument A. For example, the argument at Figure 3 includes the Attack versus Backing model, top, and then the defeat graph (⇝), bottom, of the attacking model after preferences (Figure 4).

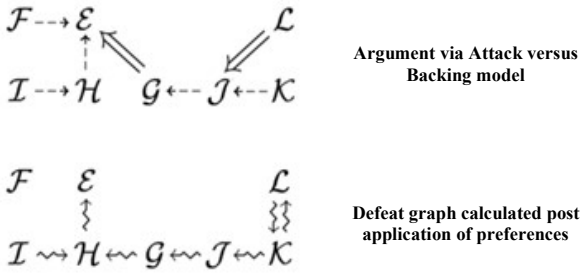

Figure 3: Pre/Post preferences

$$\preceq = \{(\mathcal{F},\mathcal{E}),(\mathcal{H},\mathcal{G}),(\mathcal{G},\mathcal{J}),(\mathcal{J},\mathcal{K})\}$$

Figure 4: Preferences

Post application of preferences (bottom Figure 3) sees claim $\mathcal{F}$ "float". $\mathcal{F}$ "floats" because, while it attacks $\mathcal{E}$, $\mathcal{E}$ is preferred over $\mathcal{F}$, per the preference relationships depicted in Figure 4.

Cohen et al. then go onto discuss primary and implicit defeats to create further distance between the two graphs. Therefore, the graph of the argument (since it focuses on attacking) will have a different structure post application of preferences and primary versus implicit defeats. The preference relationship otherwise acts as a weighted mechanism.

3.2 GRL interprets BUAF

The contribution relationships of GRL apply weights to graph edges to act as an alternative notation, thus: backers (→+) or attackers (→-). The variation in weighting (→+, →++) acts in the inductive sense to proclaim weak through strong backing relations. Depicting a weighted attacking relation is easy (→-, →--). The null or don't-know relationship is also possible (→).

The GRL interpretation of the non-monotonic logic suggested by Cohen et al., of the basic relationship between attackers (U or →-) and backers (B or →+), is intuitive as per Figure 5.

Figure 5: BUAF U/B as GRL contributions

Attack in BUAF, in fact, has three representation modes depicted in Figure 6. When looking at the combination of the base notation and then the side effect of preferences, not to mention attack relations (ℝ), the proposal from

Cohen et al. loses out it its practical application for the generalist Argumentor.

$$\mathcal{G} \dashleftarrow \mathcal{J}, \mathcal{G} \leftsquigarrow \mathcal{J}, \mathbb{R}_{b2} = \{(\mathcal{J}, \mathcal{G})\}$$

Figure 6: Attack representation modes

To help declutter, consider that attack relations ($\mathbb{R}$) are really a means to note the significant claims overall, since $\mathcal{E}$ of Figure 7 is the top claim. The intent is to declare that the principle attack relations are the rebuttal of $\mathcal{E}$ by $\mathcal{F}$ and the undermining of the backing $\mathcal{G}$ by $\mathcal{J}$, or:

$$\mathbb{R} = \{(\mathcal{F}, \mathcal{E}), (\mathcal{J}, \mathcal{G})\}$$

This rigor is necessary for the AI algorithms (especially if the graph is vast). Attack relations otherwise relate to the idea of decoration in the graphical sense, or at least of metadata on the relations[9]. The multiple cuts at the argument mean, however, as a working notation for the Designer, the review comprehensibility of the argument in BUAF thus suffers.

A simpler notational form, showing the summation of the intended relations might, however, be more useful to visually show the net intent of the argument. Weighting the contribution relation in GRL achieves an economy of notational style. Consider the preferences of Figure 4. Combined with Figure 3, for example, where $\mathcal{G}$ is preferred over $\mathcal{H}$, it results in Figure 7.

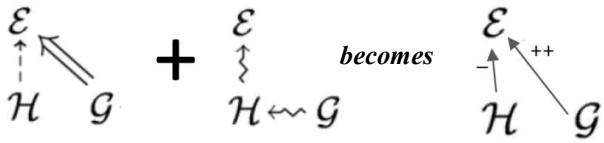

Figure 7: Net argument in GRL contributions

The GRL contribution style representation of Figure 7 shows the net result of the attack by $\mathcal{H}$ on $\mathcal{E}$, the backing of $\mathcal{E}$ by $\mathcal{G}$, and the preference of $\mathcal{G}$ over $\mathcal{H}$.

Looking now at the interactions between $\mathcal{L}$, $\mathcal{J}$ and $\mathcal{K}$, at Figure 8, the overloading of the various attacking representations need careful interpretations of the resultant net argument.

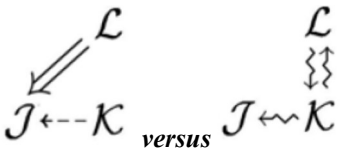

Figure 8: Argument versus defeat graph

From Figure 8, if $\mathcal{L}$ and $\mathcal{K}$ nullify one another, the influence of $\mathcal{K}$ on $\mathcal{J}$ is likely, under most circumstance, to be defunct. Nullification as contribution (right) is as per Figure 9. Adding equal backing and attacking together means the underminer $\mathcal{K}$ now has no residual influence or is null[10].

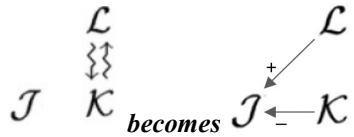

Figure 9: Nullification, $\mathcal{Null} = \mathcal{L} - |\mathcal{K}|$

Figure 4 intimates, however, that there is a residual contribution, because $\mathcal{K}$ is preferred over $\mathcal{J}$. So, despite $\mathcal{K}$ and $\mathcal{L}$ attacking one another, as $\mathcal{L} \leftsquigarrow \mathcal{K}$ are not floating Figure 10 (right) then depicts the residual Contribution of $\mathcal{K}$ versus $\mathcal{L}$. $\mathcal{L}$ is defeated as the weight of $\mathcal{K}$ is greater which shows the intent of Figure 4 in the preference for $\mathcal{K}$.

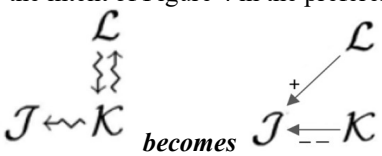

Figure 10: Residual, defeat = $|\mathcal{K}| > \mathcal{L}$

The visual decorations thus aid in the review and the comprehensibility of the graphical model as they expose the dialogue between attacking and backing. The contribution edges otherwise include meta-data to record a weighting scale ranging from -100 to 100 to support algorithms to calculate satisfaction.

The discussion on BUAF has shown the application of GRL contribution semantics, in the role of defeasible reasoning. Giorgini et al. (2004) have otherwise previously reported the formality of the application of contribution semantics, to claims of degrees of goal satisfaction (cf. Atwood et al. 2004).

4 Working example

Usefully, GSN proponents have offered GSN up in place of *i**[11] style notations when developing goal-oriented requirements (Habli et al. 2007). Unfortunately, the formalisms of GSN better represent GORN that pre-dated *i** (see Figure 11). The injection of agent formalisms into *i** is what lends it to better represent system models (Figure 12). Reasoning over the allocation of goals to actors/agents can therefore show the progression of enquiries of system attributes to closure - closure being an important property for an argument.

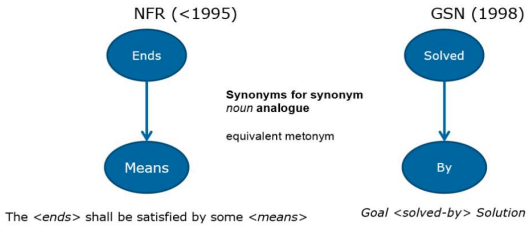

Figure 11: Core semantic similarities

[9] The use of metadata is less obtrusive, than multiple relations, since it is malleable. GRL can add metadata (including stereotypes) to elaborate on the relative importance of graph edges or nodes.

[10] $\mathcal{L}$ is also nullified, it's just a point of view. This is otherwise akin to the application of game theory. Otherwise, revision, valid objection, forfeiture, or annulment are all qualities of defeasible reasoning.

[11] Pronounced i-star and italicized by convention.

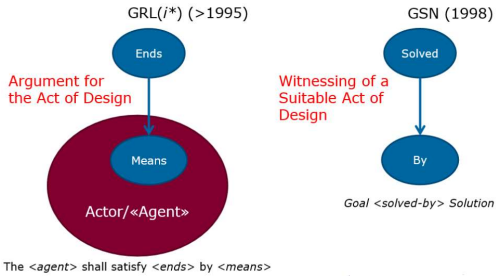

Figure 12: Agency enhances argument

GRL is a light-weight derivative of *i** (Amyot and Mussbacher, 2011) that allows modification of the semantics by use of «stereotyping», which is a familiar tool of system engineers by way of SysML/UML. To that end GSN proponents can apply their reasoning idioms by way of mapping GSN to GRL per Figure 13.

GRL	GSN
Actor/«Agent»	NIL STOCKS!!
Goal	Goal
Task	«Strategy»
Resource	«Solution»
Belief (along with Belief Link)	«Justification», «Context», «Assumption»
Dependency (link)	Solved-By (link)

Figure 13: GSN to GRL mapping

To investigate the advantage of the contribution semantic of GRL, in providing defeasible reasoning (especially in association with agent formalisms), it makes sense to look at a design reasoning approach where GSN has been utilised. This allows a direct comparison. To that end, this report will revisit the analysis procedure recommended by Habli et al.

4.1 GRL in a nutshell

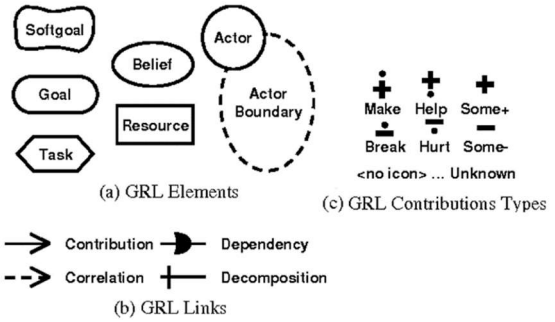

Figure 14: GRL palate

Goal and Softgoal: See heading "Goal couching and verb tense". Take goals to be quantitative and soft-goals qualitative. One can also take the stand that the difference can visually represent inductively strong and weak claims respectively. Stereotyping of goals with «Claim» for CAEesque reasoning.

Tasks: Used in the same vein as in hierarchical task analysis. Can be stereotype to «Argument» or «Strategy» for CAE/GSNesque reasoning.

Resource: Something produced and consumed. Can be stereotype to «Evidence» or «Solution» for CAE/GSNesque reasoning.

Actor (Agent): Removing *i** Agents and Roles helped to make GRL a lightweight notation. Stereotyping Actors can resuscitate «Agents» and «Roles» as needed. In this context take Actors as event driven and Agents as cognitive. The reductionism of Actor also appeals to UML and SysML modellers, since UML and SysML opt for Actors. However, Actors in UML/SysML are external to the system. The system and its environment are Actors or Agents in goal-oriented modelling.

Links: As GRL is a light-weight interpretation of *i**, the semantics are less restrictive. In *i** one would decompose goals and use contribution and dependency to model satisfaction of soft-goals. GRL has less restrictions on the combinations of intentional elements with relationship (or link) types. This is what makes GRL more malleable to the application of alternative heuristics and has also gone towards making GRL more approachable. The relationship links of interest are:

- ***Contribution*:** Satisfaction modelling offering inductive and defeasible traits. Contribution links are the analogue of Justification. In Adlard's CAE notation that would be any of: *Supports*; *Is sub-claim of*; *Is evidence for*.
- ***Dependency*:** Literally depends upon. The solid D points to the dependee from the depender. When the dependum is a resource, it stalls interface and architectural decisions to allow modelling what, not how. Dependency links are also the meta-semantic for GSN "solved-by", they certainly point in the same direction.
- ***Decomposition*:** As in part-whole decomposition. AND/OR/XOR combinational logic.
- ***Correlation*:** A correlation is an unintended side-effect.

Contribution types: These are the visual "weights" of +ve and -ve contribution that support comprehension of the net satisfaction of the goal.

4.2 GSN as a GORN

Habli et al. adopt the Quality Attribute Scenarios (QAS) model, by way of arguing their goal-orientedness, to then argue that using a general argumentation notation (GSN) is preferable to reasoning with GORN. Figure 15 depicts a QAS, which is essentially a causal chain.

More often, scenario tables capture the QAS. In the QAS table, accounting for the causal thread (see Figure 16) is by reading the table top to bottom. Of note, obscured is the involvement of agency - represented by Source (Stimulus) and Artefact (Response).

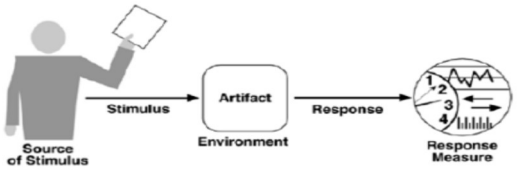

Figure 15: Quality Attribute Scenario

Portion of Scenario	Possible Value
Stimulus	Inadvertent deployment of reverser
Source	Aircraft engine
Trigger	Invalid airframe data; airframe data transmission loss; engine commission failure
Environment	Airframe is in air
End effect	Physical damage to the engine; loss of controlled flight
Effect measure	Frequency: probable, Severity: catastrophic/critical

Figure 16: QAS Tablature

Interestingly Habli et al. nominate work in GORE on "anti-goals", which is also then incorporated into the work of Wu (2007). The subtlety missed is that the conflict between +ve and –ve GRL contributions can account for Goal/Anti-Goal pairing – since it is simply a matter of opposition.

Moreover, the black and white stance of Goal/Anti-Goal struggles to consider the likelihood of two "good" goals competing, as opposed to a diametric opposition. Leaning simply upon diametrically opposed goals will likely fail the non-monotonic reasoning test in any event, certainly in the context of BUAF.

The inclusion of Anti-Goals vouches, somewhat then, for the contribution relation for *i** style notations. This is because the contribution syntax of GRL can simply treat goals as full scale contributions[12]. This despite the attempt by Habli et al. to discount GORN - and likely then GORE.

So, as discussed previously, there appears some vacillation by Argumentors on whether GORN is argument or is not argument, based on the question of verb tense. This vacillation most prevalent when Habli et al. offer GSN as an (aspirational) requirement notation by explaining GSN is a general argumentation notation, contrary to original claims that GSN is specifically a safety case argumentation notation (Kelly, 1998, p62).

4.3 QAS in GRL

Taking on board the allocation aspect, afforded by the agent formalisms in *i** style notations, results in Figure 17. In Figure 17 the Task[13] element captures both Stimulus and Response. The dependency on Response by Stimulus is the reliance on the Response to deal with the Stimulus – that is, akin to the solved-by relationship in GSN.

The Response Measure leans on the KPI semantics incorporated into GRL – since KPI are literally response measures. The environment is a belief describing the perceived context. Meta-data, on either the Artefact or the Response, could also have captured the description of the Environment.

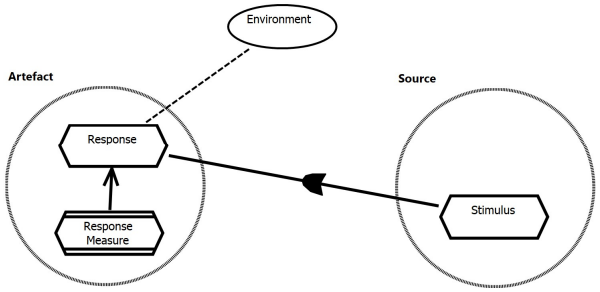

Figure 17: QAS in GRL

Figure 17 is also analogous with the meta-model for Jackson's Problem Frames which suggests a mixture of thinking patterns, as this is also (somewhat) analogous to the control-theoretic model of Leveson (Feodoroff, 2016d). Adopting this concept mixing then leads to the realization that GORN, with agent formalisms, can accommodate the net heuristics per the joint meta-model in Figure 18. So, reasoning patterns, applied to modelling in *i** style notations gives lexicon, syntax and semantics to explicitly capture concepts that relate to the safety domain (such as system models). Feodoroff (2016b, 2016d) gives a more detailed elaboration of the model at Figure 18.

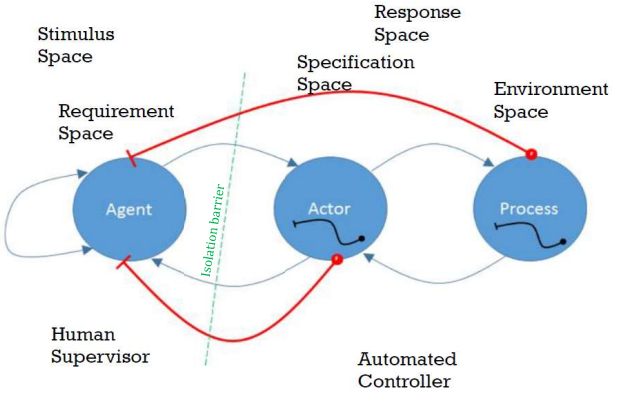

Figure 18: Combined heuristics over Agency

Taken as safety attributes, the verb-noun form can also frame the hazard/constraint antithesis pairs often promoted by safety case antagonists (Leveson 2013). That is, dependent upon the circumstance, hazards can result from a state, either or not attained, ceased, avoided or maintained, and the safety constraint (or intent) is then expressed as the antithesis.

Musing over the safety enquiry of the system using goal-oriented requirement modelling is permissible within the lexicon/syntax and semantics of GRL. As GRL has more formalisms (Part-Whole, Dependency and Agency) it certainly addresses a key design criterion for an argumentation notation (Kelly, 1998, p. 63). Addresses as GRL can capture concepts that relate to the safety domain, including the hazard analysis overtop a system model (Feodoroff 2016a, Feodoroff 2016b).

Use Case Maps (UCM), the companion notation to GRL within URN, is validated for use in hazard analysis over a system model (Wu and Kelly 2006, Wu 2007, Feodoroff 2015a). In fact, the UCM notation's semantics can account for the intuition of Wu and Kelly when meandering over the fault propagation path – UCM includes fault path modelling.

Coincidently, there is a more systematic approach known as Anticipatory Failure Determination[14] Approach (AFDA) which backs the intuition of Wu and Kelly. Sunday (2014) provides a good precis of AFDA, along with extensions. The interpretation of Sunday's "Principles of Scenario Structuring" will be intuitive to UCM modellers – especially the notational style.

Indeed, UCM's companion GRL was available to Wu and Kelly when they advocated the pairing of UCM with GSN - UCM and GRL, after all, were both modelled in the same URN software tool in 2006. However, not presented at that time was any comparison of GSN with GRL.

Indeed, the QAS model is an analogue of the notion within AFDA of SEOR: Source, Effect, Object and Result. SEOR itself is also somewhat of an analogue of Ericson's HE(source), IM(effect), T/T(object/result).

[12] Goal=100, Anti-Goal=-100

[13] Goals could just as easily be used.

[14] AFDA has roots in research and application back to 1960s. Prior to 1993 it was known as Subversion Analysis.

4.4 Replacing GSN with GRL

Starting with the example used by Habli et al., the approach will be:

1. Model the QAS graphically (instead of by tablature).
2. Add (so-called) Anti-Goals.
3. Include Anti-Anti-Goals to capture the mitigations for the Anti-Goals. This is just using defeasible reasoning.
4. Introduce a goal satisficing argument over a tree (a familiar structure to proponents of GSN), by using the QAS reasoning fragments.

A cursory pass over the paper by Habli et al. will prepare the reader of this section. The artefacts considered by Habli et al. include:

- Wheel Breaking System
- Spoiler, and
- Reverse Thrust.

Using the QAS idiom depicted at Figure 17, the first cut of the QAS for Wheel Braking Request is per Figure 19. In this first evolution, the –ve Response Measure, being the failure rate, acts to defeat the +ve Response. That visual allocation of a goal to an agent is in-line with idioms used in system engineering. Note Figure 19 thus reads visually:

> "The <*artefact*> shall <*respond*> upon <*stimulus*> when <*context*>"

which becomes:

> "The <*WBS*> shall <*engage all wheel brakes*> upon <*wheel braking request*> when <*aircraft is on ground and in landing/taking off/RTO flight phase*>"

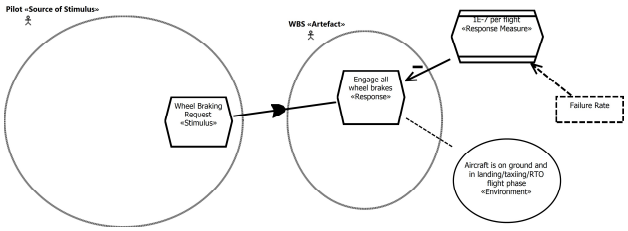

Figure 19: Wheel Braking Request

Following the proposal by Habli et al, one introduces the diametrically opposed Anti-Goal. Per Figure 20, the quirk now is that the Response Measure is a +ve influence on the Anti-Goal. The Anti-Goal takes over as the –ve influence on the Response.

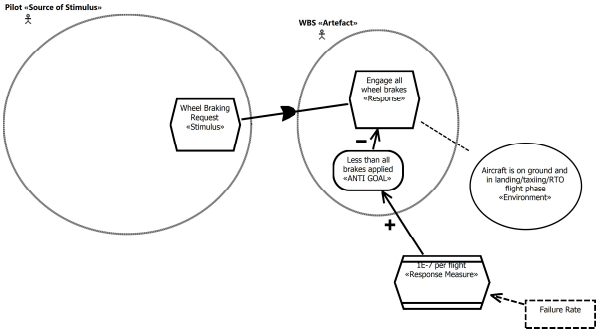

Figure 20: Wheel Braking Request revisited

Per Figure 21, the opportunity now is to capture an Anti-Anti-Goal. The arrow points at the injection of a Design Decision (decorated with a GSNesque «Solution»). As this is a toy example, consider it is permissible here to be more elaborate in the argument.

This otherwise falls under the discussion under Heading 3. So, in effect, over Figures 19..21 is the witnessing of the evolution of the dialogue, describing the non-monotonic reasoning, via the progressing of the lexical and syntactic process. The dialogue is redeemable during later review of the model. This is in contrast with the fulminatory monologue offered by GSN style notations.

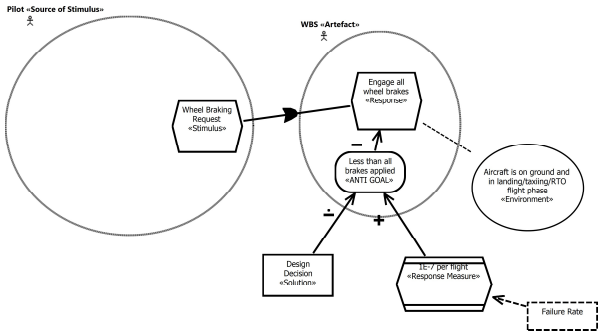

Figure 21: The Backer is countered by an Attacker

As the strategy for the argumentation base upon QAS, it is likely plausible then that the reader can see the strategy in the graph at Figure 22. That is, there is no effect on comprehension as there is no information transform. Note the grouping of the models for the three artefacts under a single safety goal. Also note that the use of part-whole decomposition within the graph, and even of dependency, can be art. That is, if it makes more sense to use contribution relations throughout, that is the prerogative of the Argumentor.

Kelly (2011, slide 37) emphasized the art by recommending not using inductive ACP on every graph edge, suggesting one need not overstate the confidence argument. Epistemologically, this is simply the idea that your argument will span the epistemic standards. This art is equally clear when using various forms of reasoning in a safety case. For example, one can use formal methods, Fault Tree Analysis (FTA) and BBN (in other than the capacity to overstate the ACP) – which all tend to span various degrees of epistemic standards than the semantics of GSN would allow. The discussion in the argumentation literature, around use of defeasible reasoning, is also trying to address higher epistemic standards than the semantics of GSN would allow.

Note compared to Figure 22, the GSN equivalent argument graph takes two separate models, after the transformation from the QAS reasoning into the GSN via information lossy and undocumented transforms. Note also, in the GSN based approach, the split of Goal versus Anti-Goal descriptions automatically challenges comprehension. The split is also then akin to the anti-pattern of "more structure" that Duan et al. (2014) warn against when authors split assurance and confidence arguments. There is therefore an economy of argument using the GRL approach.

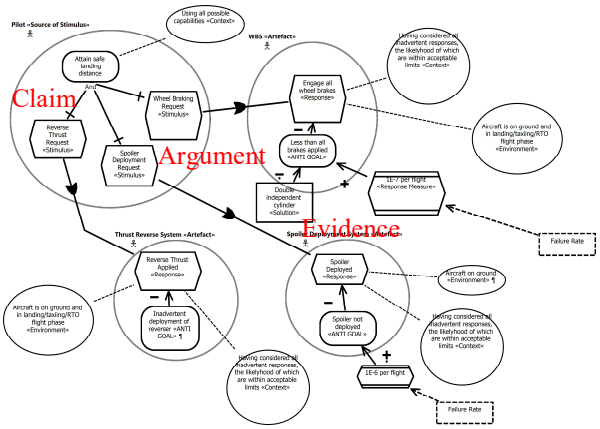

Figure 22: Agent Oriented Assurance Argument

Note the interpretation of the model in Figure 22 using the abstraction hierarchy of Claim-Argument-Evidence. Claim-Argument-Evidence is the overarching requirement, for an argument hierarchy, per IEEE/IEC/ISO 15026. As per Kelly (1998), elements of an argumentation notation, interpreted via equivocation over couching of terms, can align with the Claim-Argument-Evidence abstraction hierarchy. For example, in GSN Goals are Claims, Strategies are Arguments, and Solutions are Evidence. Indeed, mapping GRL to either of GSN or Adelarde's CAE notation via stereotyping, means GRL can also map to SACM(ARM) (Feodoroff, 2016c and 2016e).

Per Graydon and Holloway (2015) one needs strive for economy of, versus the penchant for, argument. This striving for economy is the goal both Kelly (2011, slide 37) and Duan et al. are targeting when dealing with the penchant for over-argumentation. To that end the use of defeasible reasoning in GRL, during the design process, acts in an agile fashion to 1) capture assurance argumentation during the act of design, because the Designer is empowered to do so, and 2) removes the uneconomical reverse engineering of rationale incurred where a third party non-associated Argumentor labours to retrospectively "witness" the act of design.

Indeed, codification of the Designer's knowledge must exist before it is accessible to the Argumentor. Certainly, since tacit knowledge is an epistemic mode, the tacit knowledge of the Designer will always be epistemically unknown to the Argumentor[15]. The Argumentor cannot then claim a belief in what the Designer knows because of the intellectual, informational and temporal distance of the Argumentor from the act of design.

Other design reasoning approaches are possible when using GRL because of its lightweight semantics. Figure 24, for example, melds Problem Frames, Dependability and Security modelling and heuristics from Leveson's STPA into a concept mixture that still has a non-monotonic reasoning sense (Feodoroff, 2016d). Again, the argument can be as involved as necessary. This leads to specification clauses for design decisions of the form:

> "To address the attribute of <*Attribute*>, the <*Agent|Actor*> shall provide <*Means*> by application of <*Requirement, Feature, Aspect, Tactic*> in defence of <*Threat*>"

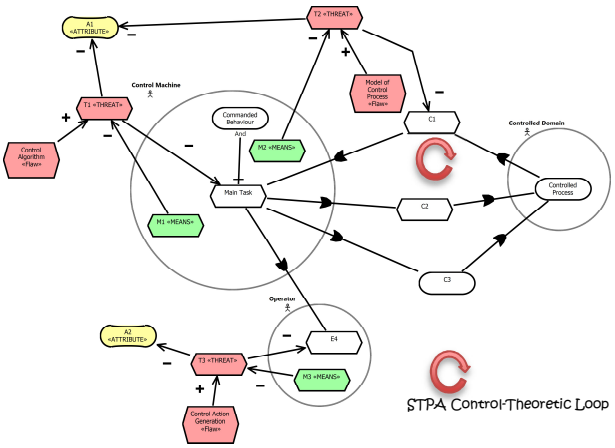

Figure 24: Graph of saplings, rather than a tree-based argument

An unfortunate analogy by Kelly (2011, slide 26), that "safety cases are bags of assertions" warrants examination. Some of the incomprehensibility of monolithic tree-based arguments as likely comes from the persistent urge to craft an enormous tree. Taking the mathematical interpretation of "bag", which is an unordered set; this does not reflect the intent of the tree structure. However, if large tree structures lead to problems with comprehension of arguments, then would a forest of saplings help or hinder?

Kelly (1998) discusses the notion of basis when context elements where introduced into GSN. Indeed, assertions and justifications are also basis. To that end, GRL beliefs provide basis. So, couching GRL beliefs as context, assumption or justification is possible since the definitions of context, assumption and justification use belief as a basis. Basis also comes into play in the computer science notion of inductively defined sets which include:

- Basis
- Inductive rule
- Closure

The allocation of goals to actors/agents in GORE, as mentioned previously, is an act of closure. As GORN include semantics for basis, and if we stretch the idea of inductive rules to non-monotonic reasoning rules, especially when applying normative heuristics, then we might opt to think of the argument as a forest of saplings. To this end we look at a safety case as an ordered-bag of non-monotonically refined reasons (saplings per Figure 24). The repeating of the keys to an ordered-bag makes the notion of an ordered-bag appealing, it also keeps any appeal of Kelly's bag analogy.

The appeal comes because it addresses the problem caused by the ALARP pattern. The safety case pattern for ALARP is an equivocation of the notion of a dependability assurance case. This confounding is because the un-evolved argument, to justify the residual risk, should be in terms of the trade-offs between safety and the other quality attributes of the system. From the model of Avižienis et

[15] The difference between Designer and Argumentor appeals to 1) Kelly's dichotomy of Rationale versus Argument, and 2) the industry trend to use third party, non-associative safety case authors.

al. (2004), take dependability as made up of the following attributes:

- Availability
- Reliability
- Safety
- Confidentiality
- Integrity
- Maintainability

If the dependability attributes are the keys into the ordered-bag of non-monotonically refined reasoning over all the attributes, then selecting the safety key would return all the results of the safety enquiries made over the system model. This would include places touched by trade-off[16].

This is a somewhat different stance to the lay understanding of the difference between a safety case and a dependability case – being there may be a legal requirement for a safety case. That is, there should be no difference in terms of application of non-monotonic reasoning, epistemology, nor of epistemic standards – though a safety case might lean towards a greater proportion of higher epistemic standards[17].

5 Of epistemic standards

As intimated, the sense of art, when applying confidence, is really about the stringency of epistemic standards applied to an argument fragment. Epistemic low standards are set when concurrence exists between participants either side of the dialogue.

Expressing concurrence might be in the use of GSN to argue compliance arguments. Surely application of epistemic low standards is okay when claiming compliance? The chapter and verse are set from above! Certainly, transformation of regulatory standards into GSN models appeals (Hall-May and Kelly, 2011).

There is, however, a problem with the denouncement of checklists in that process. Both GSN semantics and checklists can be characterised as un-weighted trees, so there is no vast difference in the semantics of the two approaches. This is especially true where both the checklist and the GSN argument are based upon the structure provided by the regulatory standard. The basis is the structure and so there is otherwise no discriminator between tablature and tree-based representations.

Reaching compliance is a per chapter summation of the compliances of the verses per chapter. Showing compliance with the verse will depend on the confidence in the satisficing supplied by the evidence. It is necessary then to communicate the weight of the compliance of the verse and likely also any exceptions. This leads to the need for higher epistemic standards and likely then reasoning non-monotonic in nature. Thus, in GRL the weighted contributions of KPI, and the challenges by exceptions, mean compliance modelling in GRL supports our non-monotonic reasoning goal. The GRL KPI mechanism thus offers a discriminator between tablature versus tree-based compliance assessment.

The KPI syntax and semantics of GRL comes from work in the regulatory compliance assessment space (Pourshahid et al. 2009, Badreddin et al. 2013). This is a space, again, where GSN proponents have offered GSN as a preference over GORE (Hall-May and Kelly, 2011). The important points are that: KPI in GRL gives a weighted assessment of compliance; proposals exist to use GRL in forward engineering of laws and regulations (Braun et al. 2012); and authors are now investigating use of GRL in the regulatory management space (Akhigbe et al., 2017).

With reference to the notion of fulminatory specifications discussed above. The fulminatory nature of GSN was because of the absence of the semantic mechanisms to capture scepticism. GSN has optionality semantics, which might pass as OR or OR-NOT but will not act as scepticism. The design of optionality semantics was to allow capturing of argument pattern abstractions and generalisations for re-use, but not to turn up in the final argument. GSN arguments themselves are thereby weighty in the use of non-optional unweighted solved-by relations.

GSN's resultant combinational logic over solved-by relations is (silent) AND – quirkily half the AND/OR combination logic Kelly (1998, p. 65) denounces. The arrows on the solved-by relation lead the eye to the leaf solutions of the tree, which is the same direction as the Part-Whole decomposition. Taking that association to heart, one notes that requirement and system representations, that include decomposition, also embrace a fulminatory nature (see Figure 25) – intimating a belief in the requirement for application of epistemic low standards, or the capturing of reasoning in some other view.

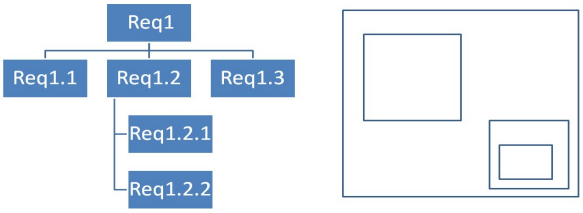

Figure 25: Fulminatory Specifications

Whether or not non-monotonic reasoning semantics are relevant to a specification view is likely the seed of debate. SysML, for example, has a raft of relationships already, including: containment, trace, derived requirement, satisfy, verify and refine. Use of "satisfy" is very narrow in SysML as it occurs when mapping the somewhat equivalent function, within a COTS solution, to a target system requirement. There is no weighting of those SysML relationships per se, so they are again fulminatory in nature. This is especially true as there is no provision for semantics of attack. The addition of a weighted contribution relationship would likely be the least invasive approach.

Atwood et al. (2004) investigated the notion of "rich traceability", as proposed by Dick (2002), to justify requirements traceability. However, the model proposed by Dick is simply a basic GORN. Much like GSN, Dick's model is a subset of GRL semantics. Similar also to GSN,

[16] It is also this necessity for trade-off between quality attributes that overturns the usefulness of the notion of Anti-Goals – as discussed previously.

[17] Since defeasible reasoning also finds its fullest expression in jurisprudence.

Dick's "rich traceability" is sans non-monotonic reasoning semantics. As the argumentation community deposes "rich traceability" with GSN, goals persist within the typed argument framework. Therefore, a choice is to add goals to SysML. Goals, within a different semantic model, already sit in the data model for DoDAF. DoDAF is already a profile atop SysML.

Epistemic reasoning itself is agent oriented, something illuminated in the argumentation literature. Expressing the act of (T)hinking, (B)elieving and (K)nowing[18], is where Agent C reasons by one of the following:

$$T_c A \vdash A$$
$$B_c A \vdash A$$
$$K_c A \vdash A$$

Whether this is an inroad to Agents within SysML, in the cognitive science sense, and as applied in GORE, is another point for debate. Hall-May and Kelly (2011) advocate agent-oriented thinking in some argument contexts. Though, without the aid of agent formalisms, n-stove-piped-goal-trees are the actual outcome. The approach by Hall-May and Kelly is otherwise a compliance argumentation approach. As discussed previously, compliance is the meat of KPI modelling adaptations in GRL. Compliance modelling in GRL being the undercutting rebuttal, to the intuition of Hall-May and Kelly, when declaring that GORN cannot systematically model regulatory compliance.

Certainly, rejection of agency has otherwise occurred in the AADL RDAL annex (Blouin, 2013). Incorporation of UCM into RDAL has occurred while ignoring GRL. Replacement of GRL was with a goal notation stunted semantically back to goals decomposed by AND/OR, and not dissimilar to "rich traceability". Therefore, RDAL opts out of both socio-technical modelling and the inclusion of non-monotonic reasoning.

This likely speaks to the resistance of the "main stream" to invest in the cognitive science, and likely the rudiments of sophisticated argument. Interesting because cognitive science is from whence accident models come, and accident models are predominantly socio-technical.

Socio-technical systems are the meat of trade for GORE because of its roots in cognitive science. It might then be the standoffishness of the "main stream", to cognitive sciences, that is the real reason safety is not well integrated into systems engineering. Is cognitive science more left of centre than safety case boffins and their philosophical debates about safety case argumentation? GSN is, of course, sans agent formalisms.

The notion of referring to a GORN as "rich traceability" appeals then as it uses a "hip" term to avoid the connotation of cognitive science. This is analogous to efforts at Amazon to encourage software engineers to use formal methods (Newcombe, 2014). Amazon simply avoided using formal method terminology and referred to TLA+ as "exhaustively tested pseudo code".

It strikes one then that rather than referring to GRL as: a goal-oriented requirement notation, that provides a typed argumentation framework which addresses reasoning over a range of epistemic standards, applicable when drafting argumentation supportive of system assurance claims, by use of both fulminatory and non-monotonic semantics; one could just use the "hip" phrase "GRL provides Rich Explanation™". Towards that goal, Amyot et al. (2016) have begun brainstorming the integration of URN with SysML.

The real potential advantage is likely when layering enterprise models, such as TOGAF or DoDAF, over the top of a suitably modified SysML. This has the potential of treating the assurance argumentation as an integrated part of models within viewpoints – rather than as a separate and un-integrated artefact.

For example, since DoDAF viewpoints manage the context of the enterprise concerns, the viewpoints' models might also present the assurance reasoning, framed by the context of those viewpoints' models (Feodoroff, 2016e). This is because the viewpoints also act as keys into the ordered-bag analogy. With the forested saplings of reasoning throughout, console-based queries of the model; keyed on enterprise dependability attributes and returning into the variable X; might look like the following:

```
(ViewpointID(ModelID, safety(X))) ; model level
(ViewpointID(_,safety(X))) ; viewpoint level
(_(_,safety(X))) ; safety case
```

As TOGAF/DoDAF viewpoints tend to have a temporal context, aspirational through to qualification, each viewpoint would set the system safety context of the following system phase via the viewpoints (Feodoroff, 2016e). That addresses Kelly's proposal for early declarations for the aspirations of the safety case.

The Standards viewpoint of DoDAF is currently passive as it only lists standards and predicts emergence of standards. The use of GRL (with KPI modelling) could evolve the Standards viewpoint into a Standards Compliance viewpoint – which is one third of the trilogy of safety case viewpoints: Risk, Compliance and Confidence (a.k.a. Qualification).

Otherwise, if using TOGAF/DoDAF viewpoints and an Agent Oriented argument, this addresses Kelly's concerns for safety case impact analysis. Addressed because the change impact on the requirements and design would directly challenge the assurance reasoning embedded in each viewpoint-model affected by change.

Ultimately, epistemically lucky is the observation that inclusion of graphical goal-oriented reasoning, within the enterprise model, would also capture the intent of Leveson's Intent Specifications – and in a notation sympathetic to the semantics demanded by IEEE/IEC/ISO 15026 for assurance arguments!

Note, however, the overloading of the term Confidence. In this paper confidence is firstly in the weighting of the edges of the argument graph as they go towards the top goal. This is an aspiration for proponents both of GSN (cf. ACP) and those of GRL. Moreover, it is also the achievement of the transparency of the process, which aims to prove the countering of scepticism, through the dialogue captured by the non-monotonic reasoning.

[18] Debate rages in the philosophy about the trip point between belief and knowledge.

The Confidence facet of the safety case trilogy came, however, from the world of software reliability – which is why it is an equivocation of Qualification[19]. This is also the point that confounds Confidence and Compliance, since if one interprets software (or system) reliability as the basis for your confidence, this would result in the planned inclusion of software (or system) reliability in a V&V process. The definitions for Verification and Validation being Kelly's explanation of the two parts of safety case Confidence: Trustworthiness and Appropriateness.

Compliance can be misleading also as it is simply the external demand for evidence of conformity to a model. In many cases, compliance in the safety case literature likely is only meant to mean conformity. The feeling that Compliance is not safety means that Compliance may limit the span of conformity applied during safety enquiries.

Confidence then is the epistemological belief gotten through the methodological nature of the V&V process. Third party ISO 9001 and CMMI auditing, and then certification by a third party, all amplify the confidence in the span of conformity reached in the processes. Unless one really can gain added (so-called) confidence, by modelling the entire V&V process and its outcomes, with a fulminatory argumentation notation, and it not simply become a second unweighted compliance/conformity argument? In any event, without the process there is no basis for confidence – no matter how large the argument.

As the fulminatory nature of argumentation notations sees reasoning about safety outside the reasoning for design, it likely also then pushes safety outside the V&V processes. On the other hand, non-monotonic reasoning about assurance goals, within the requirements and design notations, would integrate V&V outcomes into embedded preliminary and architectural safety claims.

6 Conclusion

Discounted as rationale and not argument, system engineers have thus overlooked goal-oriented requirement notations as the means to express assurance claims directly during the act of system design. The need for making assurance claims, created by the requirement for certification, passed then to Argumentors and the industry of Argumentation. The narrow view that rationale and argument are somehow different, however, appeals only to the temporal aspect, where reasoning early in a system lifecycle (rationale) is aspirational and therefore not arguing (instant) achievement of goals per se.

The notion of phased safety cases cannot distance itself, however, from aspirations for system safety expressed in the early high-level goals or claims. So, the malleability of the temporal aspect of goal couching means there is a place for goal-oriented requirement notations in place of the more general argumentation notations.

Goal oriented requirement notations achieve this through capturing defeasible reasoning during the design process. If the enquiry of the system is safety related, then the reasoning over the lexical and syntactic process can justify the progression of assurance from aspiration to confidence reached through system qualification.

The GORE interpretation of goals can act both in the imperative and past participle sense. Given the propensity for use of goal-based notations to present assurance arguments, this eases the migration to GORN. This ease is because the semantics of the argumentation notations are a sub-set of GRL. So, both Designers and Argumentors can have the same notation and language.

7 References

Akhigbe, O., Heap, S., Amyot, D., Richards, J.S. (2017): Exploiting IBM Watson Analytics to Visualize and Analyze Data from Goal-Based Conceptual Models. *Proc. 36th International Conference on Conceptual Modelling 2017*, Valencia, Spain, CEUR Workshop Proceedings, Vol-1079, pp. 352-355.

Amyot, D., Anda, A.A., Baslyman, M., Lessard, L., Bruel, J-M. (2016): Towards Improved Requirements Engineering with SysML and the User Requirements Notation. Proc. *IEEE 24th International Requirements Engineering Conference 2016*, Beijing, China, pp. 329-334.

Amyot, D. and Mussbacher, G. (2011): User Requirements Notation - The First Ten Years, The Next Ten Years. Invited paper, *Journal of Software, Vol. 6, No. 5*, Academy Publisher, May 2011, pp. 747-768.

Armstrong, J. and Paynter, S. (2004): The Deconstruction of Safety Arguments Through Adversarial Counter-argument. Technical Report *Series CS-TR-832*, University of Newcastle upon Tyne.

Atwood, K., Kelly, T., McDermid, J. (2004): The use of satisfaction arguments for traceability in requirements reuse for system families. Position Paper. *Proc. Int. Workshop on Requirement Reuse in System Family Engineering, 8th Int. Conf. on Software Reuse 2004*, Carlos III University of Madrid, Spain, pp. 18-21.

Avižienis, A., Lapri, J-C., Randell, B., Landwehr, C. (2004): Basic Concepts and Taxonomy of Dependable and Secure Computing. Paper. *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, January-March 2004.

Badreddin, O., Mussbacher, G., Amyot, D., Behnam, S.A., Rashidi-Tabrizi, R., Braun E., Alhaj, M., Richards, G. (2013): Regulation-Based Dimensional Modeling for Regulatory Intelligence. *Proc. Sixth International Workshop on Requirements Engineering and Law*, Rio de Janeiro, Brazil, pp. 1-10.

Blouin, D. (2013): Modeling Languages for Requirements Engineering and Quantitative Analysis of Embedded Systems. Ph.D. thesis. University Bretagne Sud,

Braun, E., Cartwright, N., Shamsaei, A., Behnam, S.A., Richards, G., Mussbacher, G., Alhaj, M., Tawhid R. (2012): Drafting and modeling of regulations: Is it being done backwards? *Proc. Fifth International Workshop on Requirements Engineering and Law*, Chicago, IL, USA, pp 1-6.

[19] Safety case confidence in V&V appears to have been generalised from the very specific As Confident As Reasonably Practicable (ACARP). In ISO/IEEE governed processes one would as likely adopt IEEE 1633 for Software Reliability, for example, to be the basis for claiming ACARP within an IEEE 1012 V&V model.

Cohen A., Garc´ıa, A.J., Simari G.R. (2011): An Argumentation Framework with Backing and Undercutting. *Proc. International Symposium on Foundations of Information and Knowledge Systems*, Berlin, Heidelberg, Springer, pp. 107-123.

Dick, J. (2002): Rich Traceability. Proc. *1st International Workshop on Traceability for Emerging Forms of Software Engineering*, Edinburgh, UK, 2002.

Duan, L, Rayadurgam, S, Heimdahl, M.P.E, Ayoub, A, Sokolsky, O, Lee, I (2014): Reasoning About Confidence and Uncertainty in Assurance Cases: A Survey. *Proc. FHIES 2014 and SEHC 2014*, Springer, pp. 64-80.

Dung, P.M. (1995): On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2). pp. 321–358.

Feodoroff, R. (2015a): URN in System of Systems - Deviation modelling in the large. White Paper;
DOI: 10.13140/RG.2.1.4635.6968/1.

Feodoroff, R. (2016a): Resilient URN - FRAM. Submitted Manuscript;
DOI: 10.13140/RG.2.2.35764.65929/7.

Feodoroff, R. (2016b): Resilient URN – STPA. Submitted Manuscript;
DOI: 10.13140/RG.2.2.23191.57765/1.

Feodoroff, R (2016c): URN in place of GSN - Design Rationale versus Assurance Argument. Presentation. *Proc. Australian System Safety Conference 2016*;
DOI: 10.13140/RG.2.1.1175.8960.

Feodoroff, R (2016d): URN in place of GSN - Design Rationale versus Assurance Argument. Preprint. *Proc. Australian System Safety Conference 2016*;
DOI: 10.13140/RG.2.1.1378.6480.

Feodoroff, R. (2016e): Intentional Enterprise Architecture. *Proc. IEEE Systems Conference 2016*, Orlando, Florida, USA;
DOI: 10.1109/SYSCON.2016.7490555.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R. (2002): Formal Reasoning Techniques for Goal Models. *Journal on Data Semantics*, LNCS, Vol. 2800, Springer, pp. 1-20.

Graydon, P.J. and Holloway, C.M. (2015): "Evidence" Under a Magnifying Glass: Thoughts on Safety Argument Epistemology. *Proc. 10th IET System Safety and Cyber-Security Conference 2015*, Bristol, UK, IET, pp. 24-29.

Greenwell, W.S., Knight, J.C., Holloway, C.M., Pease, (2006): A Taxonomy of Fallacies in System Safety Arguments. *Proc. 24th International System Safety Conference 2006*, Albuquerque, NM, USA, NASA Langley Research Center.

Goodenough, J.B., Weinstock, C.B., Klein, A.Z. (2015): Eliminative Argumentation: A Basis for Arguing Confidence in System Properties. Technical Report CMU/SEI-2015-TR-005, Software Engineering Institute, Carnegie Mellon University.

Habli, I., Wu, W., Attwood, K., Kelly, T. (2007): Extending Argumentation to Goal-Oriented Requirement Engineering. *Proc. International Conference on Conceptual Modeling 2007*, Auckland, New Zealand, pp. 306-316.

Hawkins, R., Kelly, T, Knight, J., Graydon, P. (2011): A New Approach to Creating Clear Safety Arguments. *Proc. Nineteenth Safety-Critical Systems Symposium 2011*, Southampton, UK, Springer, pp. 3-23.

Holloway, C.M. and Johnson, C.W. (2009): Towards a comprehensive consideration of epistemic questions in software system safety. *Proc. 4th IET International Conference on System Safety 2009*, London, UK, IET, pp. 105-109.

Kelly, T. (1998): Arguing Safety: A systematic approach to managing safety cases. Ph.D. thesis. University of York, York, UK.

Kelly, T. (2011): Relating Risk and Confidence - A Structured Approach to Constructing Assurance Cases. Tutorial. *Summer Software Symposium on Assurance Cases: New Techniques and New Guidance 2011*, Software Engineering Center, University of Minnesota, Minnesota, USA. Accessed: https://www.umsec.umn.edu/events/SSS-2011.

Leveson, N. (2013): An STPA Primer. Technical Report, MIT, Cambridge, Massachusett, USA.

Newcombe, C. (2014): Why Amazon Chose TLA+. *Lecture Notes in Computer Science*, vol. 8477, pp. 25-39, 2014.

Pourshahid, A. Amyot, D., Peyton, L., Ghanavati, S., Chen, P. Weiss, M. Forster, A.J. (2009): Business process management with the user requirements notation. *Electronic Commerce Research*, Dec 2009, Volume 9, Issue 4, pp 269–316.

Rushby, J. (2013): Logic and Epistemology in Safety Cases. *Proc. SAFECOMP 2013*, Telouse, France, Springer LNCS 8153, pp. 1-7.

Rushby, J., Xu, X., Rangarajan, M., Weaver, T.L. (2015): Understanding and Evaluating Assurance Cases. Technical Report NASA/CR–2015-218802, Boeing Research and Technology, Seattle, WA, USA.

Sunday, E. (2014): Extension and Modification of Anticipatory Failure Determination Approach Based on I-TRIZ. MSc Thesis. University of Stavanger, Stavanger, Norway.

Toulmin, S. (2003): The uses of argument. Cambridge University Press, New York.

Wu, W. and Kelly, T. (2006): Managing Architectural Design Decisions for Safety-Critical Software Systems. *Proc. Quality of Software Architectures QoSA 2006*, Berlin, Heidelberg, Springer, pp. 59-77.

Wu, W. (2007): Architectural Reasoning for Safety-Critical Software Application. Ph.D. thesis. University of York, York, UK.

Zhao, X., Zhang, D., Lu, M., Zeng F. (2012): A new approach to assessment of confidence in assurance cases. *Proc. SAFECOMP 2012*, Magdeburg, Germany, Springer, pp. 79-91.